Purrr and Broom Example

Brian High

11/1/2019

Fit multiple models using purrr and broom

See: https://r4ds.had.co.nz/many-models.html

Setup

```
# Load packages.
pacman::p_load(datasets, tibble, dplyr, tidyr, purrr, broom, modelr, knitr)

# Get the dataset.
data(mtcars)
```

Prepare data

```
# Import the dataset into a nested tibble. Split car name into make and model.
df <- as_tibble(mtcars %>%
                  mutate(model = row.names(mtcars))) %>%
  mutate(make = gsub('^(\\w+) .*$', '\\1', model),
         model = gsub('^\\w+ (.*)$', '\\1', model)) %>%
  group_by(make, model) %>% arrange(make, model)
```

View data

- Use `kable()` from the `knitr` package.

```
# Examine the dataset.
kable(df %>% head(10), format = 'markdown')
```

mpg	cyl	disp	hp	drat	wt	qsec	vs	am	gear	carb	model	make
15.2	8	304.0	150	3.15	3.435	17.30	0	0	3	2	Javelin	AMC
10.4	8	472.0	205	2.93	5.250	17.98	0	0	3	4	Fleetwood	Cadillac
13.3	8	350.0	245	3.73	3.840	15.41	0	0	3	4	Z28	Camaro
14.7	8	440.0	230	3.23	5.345	17.42	0	0	3	4	Imperial	Chrysler
22.8	4	108.0	93	3.85	2.320	18.61	1	1	4	1	710	Datsun
15.5	8	318.0	150	2.76	3.520	16.87	0	0	3	2	Challenger	Dodge
14.3	8	360.0	245	3.21	3.570	15.84	0	0	3	4	360	Duster
19.7	6	145.0	175	3.62	2.770	15.50	0	1	5	6	Dino	Ferrari
32.4	4	78.7	66	4.08	2.200	19.47	1	1	4	1	128	Fiat
27.3	4	79.0	66	4.08	1.935	18.90	1	1	4	1	X1-9	Fiat

Nest data

Subset the data by whether or not the "make" is an American car maker. Nest the groups for easier automation of modeling.

- Use `group_by()` from the `dplyr` package.
- Use `nest()` from the `tidyr` package.

```
# Identify American car makers with: usa = TRUE.
df <- df %>%
  mutate(usa = ifelse(make %in% c('Hornet', 'Valiant', 'Duster', 'Merc',
                                  'Cadillac', 'Lincoln', 'Chrysler', 'Dodge',
                                  'AMC', 'Camaro', 'Pontiac', 'Ford'),
                      TRUE, FALSE))

# Nest by 'usa'.
df <- df %>% group_by(usa) %>% nest()
```

Fit models

Fit using `lm()` with multiple formulas for each nested group.

- Use `formulas()` and `fit_with()` from the `modelr` package.
- Use `map()` from the `purrr` package.

```
# Define formulas.
lm_formulas <- formulas(~mpg,
                                ~ cyl,
                                ~ cyl + disp,
                                ~ cyl + disp + hp,
                                ~ cyl + disp + hp + wt)

# Fit models.
df <- df %>%
  mutate(model = map(.x = data, .f = ~fit_with(lm, lm_formulas, data = .x)))
```

Get model summaries and estimates

Extract the summary information from the models. We need to use `lapply()` with `map()` because we are using many models and `fit_with()` created `list` output like `lapply()` would have. So we use `lapply()` to get "inside" of the `list` output of `fit_with()`.

- Use `glance()` and `tidy()` from the `broom` package.
- Use `map()` from the `purrr` package.
- Use `lapply()` from `base` to apply `glance()` and `tidy()` to many models.

```
# Extract model summaries with glance().
df <- df %>%
  mutate(resid = map(model, ~lapply(.x, glance)))

# Extract model estimates with tidy().
df <- df %>%
  mutate(est = map(model, ~lapply(.x, tidy)))
```

View model summaries for "usa"

- Use `unnest()` from the `tidyr` package to unpack the `list` columns.
- Use `mutate()` from the `dplyr` package to include the formula names.

```
df.resid <- df %>% select(usa, resid) %>% unnest(resid) %>%
  mutate(formula = as.character(lm_formulas)) %>% unnest(resid)
```

- Use `kable()` from the `knitr` package.

```
kable(df.resid[, c(1:7, 11, 13)], format = 'markdown', digits = 4)
```

usa	r.squared	adj.r.squared	sigma	statistic	p.value	df	deviance	formula
TRUE	0.6151	0.5925	2.3588	27.1667	0.0001	2	94.5896	mpg ~ cyl
TRUE	0.6748	0.6341	2.2350	16.5985	0.0001	3	79.9227	mpg ~ cyl + disp
TRUE	0.6955	0.6346	2.2336	11.4200	0.0004	4	74.8316	mpg ~ cyl + disp + hp
TRUE	0.8012	0.7444	1.8682	14.1028	0.0001	5	48.8625	mpg ~ cyl + disp + hp + wt
FALSE	0.5342	0.4919	4.0552	12.6157	0.0045	2	180.8889	mpg ~ cyl
FALSE	0.6859	0.6231	3.4925	10.9187	0.0031	3	121.9788	mpg ~ cyl + disp
FALSE	0.7073	0.6097	3.5539	7.2489	0.0090	4	113.6744	mpg ~ cyl + disp + hp
FALSE	0.8122	0.7183	3.0193	8.6498	0.0053	5	72.9304	mpg ~ cyl + disp + hp + wt

View model estimates for "usa"

- Use `unnest()` from the `tidyr` package to unpack the `list` columns.
- Use `mutate()` from the `dplyr` package to include the formula names.

```
df.est <- df %>% select(usa, est) %>% unnest(est) %>%
  mutate(formula = as.character(lm_formulas)) %>% unnest(est)
```

- Use `kable()` from the `knitr` package.

```
kable(df.est, format = 'markdown', digits = 4)
```

usa	term	estimate	std.error	statistic	p.value	formula
TRUE	(Intercept)	31.8244	2.9240	10.8839	0.0000	mpg ~ cyl
TRUE	cyl	-2.0924	0.4014	-5.2122	0.0001	mpg ~ cyl
TRUE	(Intercept)	30.0152	2.9648	10.1238	0.0000	mpg ~ cyl + disp
TRUE	cyl	-1.2196	0.6357	-1.9185	0.0731	mpg ~ cyl + disp
TRUE	disp	-0.0147	0.0086	-1.7135	0.1059	mpg ~ cyl + disp
TRUE	(Intercept)	29.2138	3.0673	9.5243	0.0000	mpg ~ cyl + disp + hp
TRUE	cyl	-0.8418	0.7372	-1.1420	0.2714	mpg ~ cyl + disp + hp
TRUE	disp	-0.0110	0.0093	-1.1873	0.2536	mpg ~ cyl + disp + hp
TRUE	hp	-0.0178	0.0176	-1.0102	0.3284	mpg ~ cyl + disp + hp
TRUE	(Intercept)	36.7422	3.7682	9.7506	0.0000	mpg ~ cyl + disp + hp + wt
TRUE	cyl	-1.2569	0.6351	-1.9790	0.0678	mpg ~ cyl + disp + hp + wt
TRUE	disp	0.0086	0.0106	0.8087	0.4322	mpg ~ cyl + disp + hp + wt
TRUE	hp	-0.0212	0.0148	-1.4358	0.1730	mpg ~ cyl + disp + hp + wt
TRUE	wt	-2.6010	0.9535	-2.7278	0.0163	mpg ~ cyl + disp + hp + wt
FALSE	(Intercept)	40.0742	4.4366	9.0326	0.0000	mpg ~ cyl
FALSE	cyl	-3.1962	0.8999	-3.5519	0.0045	mpg ~ cyl
FALSE	(Intercept)	32.6497	5.1004	6.4014	0.0001	mpg ~ cyl + disp
FALSE	cyl	0.8070	1.9796	0.4077	0.6921	mpg ~ cyl + disp
FALSE	disp	-0.0928	0.0422	-2.1976	0.0527	mpg ~ cyl + disp
FALSE	(Intercept)	33.5426	5.3056	6.3221	0.0001	mpg ~ cyl + disp + hp
FALSE	cyl	0.8300	2.0146	0.4120	0.6900	mpg ~ cyl + disp + hp
FALSE	disp	-0.1309	0.0637	-2.0548	0.0701	mpg ~ cyl + disp + hp
FALSE	hp	0.0333	0.0411	0.8109	0.4384	mpg ~ cyl + disp + hp
FALSE	(Intercept)	41.5452	5.8861	7.0582	0.0001	mpg ~ cyl + disp + hp + wt
FALSE	cyl	0.9832	1.7131	0.5739	0.5818	mpg ~ cyl + disp + hp + wt
FALSE	disp	-0.0578	0.0643	-0.8987	0.3950	mpg ~ cyl + disp + hp + wt
FALSE	hp	0.0144	0.0360	0.3985	0.7007	mpg ~ cyl + disp + hp + wt
FALSE	wt	-6.6945	3.1666	-2.1141	0.0674	mpg ~ cyl + disp + hp + wt